MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

DCS5088 – OBJECT ORIENTED PROGRAMMING

(For DIT students only)

9 MARCH 2019
9.00 a.m – 11.00 a.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This examination paper consists of **12** pages. There are **TWO (2) sections** in this paper.

2. **SECTION A:** There are **3** structured questions.

3. **SECTION B:** There is **1** structured question.

SECTION A: Structured Questions (Total: 30 Marks)

Instruction: Please write all your answers in the Answer Booklet provided.

QUESTION 1 (10 marks)

a) Given the following code segments, identify the output of **z**.

```
int a, b, z;
a = 1;
b = 4;
while ( a < b ){
  a = 2*a;
  b = b+1;
}
  z = b;
```

[1 mark]

b) Write the program segments based on the descriptions given:

i. Create a structure called *Shoes*. The data members are:
- *size* ***(int)***
- *color* ***(character array, size 30)***

[1 mark]

ii. Declare an object name *s1* of struct *Shoes*. Set the size to 8. [2 marks]

c) Given the program below, answer the following questions:

```
#include <iostream>                                          //line1
using namespace std;                                         //line2
                                                             //line3
int size (float *z, float n)                                 //line4
{                                                            //line5
    z = *z *(1+n);                                           //line6
}                                                            //line7
void display(float n)                                        //line8
{                                                            //line9
    cout<<"The current expected size is :"<<n<<endl;         //line10
}                                                            //line11
int main()                                                   //line12
{                                                            //line13
    float Sz = 3500, i=0.08;                                 //line14
                                                             //line15
    cout<<"The given standard size is: "<<Sz<<endl;          //line16
    size(&Sz, i);                                            //line17
    display(Sz);                                             //line18
    return 0;                                                //line19
}                                                            //line20
```

Continued...

i. Identify TWO lines that contain error. Fix the errors by writing down the correct answer. [1 mark]

ii. Based on the answer in (i), write the output for this program. [1 mark]

d) Given the program shown below, answer the following questions:

```
#include <iostream>
using namespace std;
class Rectangle
{     float length, width;
    public:
        void setData(float l, float w)
        { length = l; width = w;
        }

        float CalculateArea()
        {  return length*width;  }
} R3;

int main()
{
      Rectangle R1, R2, R4;
      float p = 8, q = 6;
      R1.setData(p,q);
      R2.setData(6,4);
      R4.setData(8,10);
      R3 = R2 ;
      cout<<"Area of rectangle R1    : "<< R1.CalculateArea() <<endl;
      cout<<"Area of rectangle R2    : "<< R2.CalculateArea() <<endl;
      cout<<"Area of rectangle R3    : "<< R3.CalculateArea() <<endl;
      cout<<"Area of rectangle R4    : "<< R4.CalculateArea() <<endl;
      return 0;
}
```

i. What is the output of the program. [2 marks]

ii. Write the statements to swap the contents between ***R1*** and ***R3***. [2 marks]

Continued...

QUESTION 2 (10 Marks)

a) Given the program below:

```
#include <iostream>
using namespace std;

class Room
 {  int width, length;
    public:
      // 2 a) i. Write your answer on your answer booklet

       void setData (int L, int T)
      {   width = L;
          length = T;
      }

     friend class Building;
 } ;

class Building
{ string name;
  public:
       Building()
       {  name = "FIST Building";
          cout<<"----Buiding name : "<<name<<"-------"<<endl;
 }
     // 2 a)ii. Write your answer on your answer booklet
};

int main()
 {   Building B1;
     Room r[5];
     int i  = 0, x, y;
     while( i < 5 )
     { cout<<"Enter the tables' width and length :\n";
       cin>>x>>y;
       r[i]. setData(x,y);
       i++;
     }

     B1.findLargest(r);
  return 0;
}
```

[**Note:** Refer to sample output given below. The ***bold*** items are the inputs entered by user]

Continued...

Sample output screen

```
----Buiding name : FIST Building-------
~~Object Created~~
~~Object Created~~
~~Object Created~~
~~Object Created~~
~~Object Created~~
Enter the tables' width and length :
10 8
Enter the tables' width and length :
10 6
Enter the tables' width and length :
10 15
Enter the tables' width and length :
10 10
Enter the tables' width and length :
10 12
The table area :80
The table area :60
The table area :150
The table area :100
The table area :120
The largest area is 150
```

i. At segment labelled *'//2 a) i'*, write the constructor function that displays the output of "~~Object Created~~". [2 marks]

ii. At segment labelled *'//2 a) ii'*, write the codes to define function *findLargest(...)*. This function will receive an array of 5 *Room* objects and display the area *width* x *length*) of each object. The largest room area will be determined and displayed accordingly. [4 marks]

b) Given the program below:

```
#include<iostream>
#include<iostream>
#include<iomanip>
using namespace std;
class Athlete
{ protected:

    string name; char gender;
    int age;
```

Continued...

```
    Athlete(string n, char g, int a)
    { name = n;
      gender = g;
      age = a;
    }
};


class Gymnastic : public Athlete
{  float weight, height, bmi;
   string bmi_desc;
   public:

  //2 b) i. write your answer on your answer booklet

  void calculateBMI()
  {
     bmi = weight / (height*height);
     if (bmi<18.5)
        bmi_desc = "Thin";
     else if (bmi>18.5 && bmi<25)
        bmi_desc = "Healthy";
     else if (bmi>25 && bmi<30)
        bmi_desc = "Overweight";
     else
        bmi_desc = "Obese";
  }

  void display()
  {  cout<<"Name        : "<<name<<endl;
     cout<<"Age         : "<<age<<endl;
     cout<<"Gender      : "<<gender<<endl;
     cout<<"Weight      : "<<weight<<" kg "<<endl;
     cout<<"Height      : "<<height<<" m "<<endl;
     cout<<"Body Mass Index  : "
          <<fixed<<setprecision(2)<<bmi<<endl;
        cout<<"Description      : "<<bmi_desc<<endl;
   }
};

int main()
{  Gymnastic G("Loqman", 'M', 39, 70, 1.64);
   G.calculateBMI();
   G.display();
   return 0;
}
```

Continued...

Sample output screen

```
Name          : Loqman
Gender        : M
Age           : 39
Weight        : 70 kg
Height        : 1.64 m
Body Mass Index   : 26.03
Description       : Overweight
```

i. At segment labelled *'//2 b) i'*, write the codes to create a constructor for class *Athlete*:

- Parameterized constructor with 5 parameters to set the data members ***name, gender, age, weight*** and ***height.*** Use the proper notation to call *Athelete* class's parameterized constructor by passing the necessary variables. [2 marks]

ii. Dynamic memory allocation is not utilized at the main function. Rewrite the main function to incorporate the dynamic allocation. [2 marks]

QUESTION 3 (10 Marks)

a) Write **TWO** member functions to overload the addition `"+"` operator and subtraction `"-"` operator at the commented line *'//3 a)'* in the code. The function has one parameter which is a reference object of *Point* class. The function should return an object whose having value x and y. [4 marks]

```
#include<iostream>
#include<iomanip>
using namespace std;

class Point{
   private: double x, y;
   public:
      Point()
      {
       x=y=0.0;
      }
      Point(double n1, double n2)
      {
      x=n1;
      y=n2;
      }

      //3 a) write your answer on your answer booklet


      void display()
      {
            cout<<"x : "<<x<<endl;
            cout<<"y : "<<y<<endl<<endl;
```

Continued...

```
        }

};


int main( )
{
        Point p1 (7.0, 4.0);
        Point p2 (8.0, 2.0);
        Point p3, p4;

        p4 = p1 + p2;
        p5 = p1 - p2;

        cout<<"p4's values"<<endl;
        p4.display();
        cout<<"p5's values"<<endl;
        p5.display();

        return 0;
}
```

b) Trace and write the output produced by the program below: [4 marks]

```
#include<iostream>
using namespace std;
class Staff
{    protected:
        float salary;

     public:
        virtual void display ( float s )
        {  salary = s;
           cout<< "= Your salary is :: RM " << salary<< endl;
        }

        virtual ~ Staff ()
        { cout<< "Staff class" << endl << endl; }
};

class Manager: public Staff
{     void display( float s )
      {  salary = s;
         cout<< "This is a Manager's salary :" << salary << endl;
      }

      ~Manager()
      {  static int num=1;
        cout << "Manager " << num << " salary calculation done" <<
         endl ;
         num++;
       }
};
```

Continued...

```
int main()
{   Staff j;

    Staff *p = new Manager;
    p->display(2590);
    delete p;

    p = new Manager;
    p->display(3670);
    delete p;

    p = &j;
    p->display(1000);

   return 0;
}
```

c) Observe the program below:

```
#include <iostream>

using namespace std;

class Vehicle
{  protected: string carName;
      public:
      void setName(string n)
      { carName = n;
      }
      virtual void print_name()
      {
       cout << "Vehicle name : " << carName << "\n";
      }
};
class Car: public Vehicle
{
      public:
      void print_name()
      {
       cout << "Vehicle #Car# brand     : " << carName << "\n";
      }

      ~Car()
      { cout<< "Destroy ### " <<endl;
      }
};

int main()
{   string car[2] = {"Proton", "Perodua"};
    Vehicle *C[2];
    Car obj[2];
```

Continued...

```
    for(int i=0; i<2; i++)

    {  C[i] = &obj[i];
       obj[i].setName(car[i]);
    }

    for(int i=0; i<2; i++)
    { C[i]->print_name();
    }
  return 0;
}
```

i. Write the output when virtual keyword is included in the program. [1 mark]

ii. Write the output when virtual keyword is removed from the program [1 mark]

Continued...

SECTION B (Total: 20 Marks)

Instruction: Please write all your answers in the Answer Booklet provided.

Write a complete program that gets user input for the productivity of two premium branded shoes for the last 3 months of the year. The program will display a summary detail as below:

a) Brand – Display the brand of shoes
b) Model – Display the model of shoes
c) Type – Display the type of shoes
d) Month – Display the month for the highest productivity
e) Max Quantity Produced – Display the highest quantity according to the month.

- Declare a global constant variable for the program:
  - *SIZE* (int) : Set the constant value to 3. This value is for the size of the array *month* used in this program.

- Declare a global variable for the program:
  - *highMonth* (int) : Set the value based for the highest month of production.

- Create a class called **Shoes**:
  - Protected data members :
    - *type* : string
    - *model* : string

  - Public member functions:
    - *SetShoes(....)* : Contains two string parameters to set the *model* and *type.*
    - *SetQty( )* : Set it as a pure virtual function.

- Create a class called **PremiumShoes**[derived publicly from the class **Shoes**]
  - Private data members :
    - *brand* : string
    - *month* : string[SIZE]
    - *qty* : float[SIZE]
    - *total* : float
    - *highNum* : int
    - *month_desc* : string

  - Public member functions:
    - Parameterized constructor :
      - Contains three parameters of string type. One of the parameters is used to set the *brand* and the other two parameters will be passed to function *SetShoes(...)* to set the *model* and *type.*
      - Set the array *month* with values "Oct", "Nov" and "Dec".
      - Set the *total* to 0.
      - Set the *highNum* to 0.

Continued...

- *SetQty( ) :*
  - Get user input for *qty* array of three months.
  - Accumulate the *qty* array element in *total.*

- *Display( )*
  - Display the *brand*, *model* and *type.*

- *findMax(PremiumShoes&)*:
  - Set this function as a *friend* of the class.
  - Refer to the instructions after this to define the function.

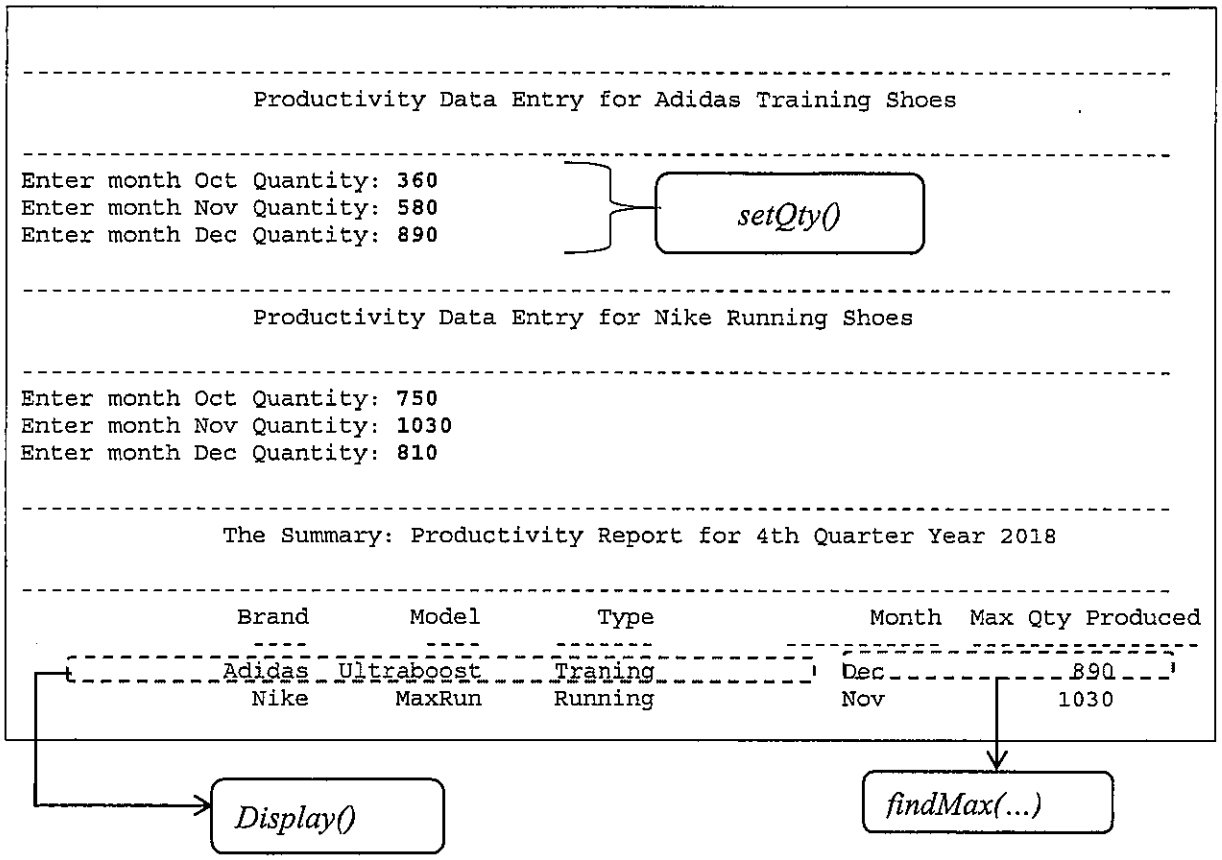
- Create a function called *findMax(PremiumShoes&)* outside the class.
  - Parameter : Reference object of **PremiumShoes** class.
  - Set the *highMonth* with the highest value of the *month* array.
  - Returns the highest value of the *qty* array. (data member of **PremiumShoes** class).

- In *main( )*:
  - Declare a pointer object, *p* of **PremiumShoes** class.
    - Use this pointer to create a new dynamic object. Pass the values "Adidas", "Ultraboost", "Training" that will set the object's *brand*, *model* and *type*.
    - Call *SetQty( ).*
  - Declare a pointer object, *q* of **PremiumShoes** class.
    - Use this pointer to create a new dynamic object. Pass the values "Nike", "MaxRun", "Running" that will set the object's *brand*, *model* and *type*.
    - Call *SetQty( ).*
  - For each dynamic object (*p* and *q*)
    - Call *Display()*
    - Display the highest quantity of the productivity by calling *findMax(....)* by passing the dynamic object using dereference operator.
  - Deallocate memory for all the dynamic objects.

[Note: Refer to sample output given below. The *bold items* are inputs entered by user.]

Continued...

Sample input and output :


Productivity Data Entry for Adidas Training Shoes
Enter month Oct Quantity: 360
Enter month Nov Quantity: 580
Enter month Dec Quantity: 890
setQty()
Productivity Data Entry for Nike Running Shoes
Enter month Oct Quantity: 750
Enter month Nov Quantity: 1030
Enter month Dec Quantity: 810
The Summary: Productivity Report for 4th Quarter Year 2018
Brand Model Type Month Max Qty Produced
Adidas Ultraboost Traning Dec 890
Nike MaxRun Running Nov 1030
Display()
findMax(...)

End of Page.